Mathematical Representations for Web Services

Dr. Joseph B. Collins
Naval Research Laboratory
4555 Overlook Ave., SW
Washington, DC 20375
202-404-7041

ABSTRACT: *Specifying at a technical level the semantic content of computational models and the services they may provide requires mathematical descriptions. Computer source code, such as C, C++, or Java, provides, at an algorithmic level, a relatively primitive form of unambiguous, mathematical specification. These computer languages are not as useful for specifying requirements, exposing assumptions, validating that designs satisfy required global properties, or for verifying that implementations conform to the design and requirements. The outward mathematical properties of software services may be documented in natural language, but they are not generally documented in machine-readable form.*

Much work has been done during the last twenty years in a variety of concurrent efforts to bring about the ability to write machine-readable, formal, representations of mathematical concepts. These may be used to represent various forms of mathematical knowledge, including mathematical specifications of software objects. We discuss how these ideas may be so applied.

1. Introduction

One might think that the epitome of clear and unambiguous descriptions is one based on mathematics. Mathematical notation itself, however, is commonly a point of contention, and there is no uniform, comprehensive standard, and hence, no unambiguous standard. Such contention is illustrated in the history of the notation for representing physical quantities with vectors [1]. This example, illustrates that poor notation, while difficult to use, can have its champions. In general, however, widely accepted standards in the representation of technical information have probably been of far greater benefit overall than might be inferred by a focus on the disputes encountered on the way to achieving those standards. Significant contention is perhaps more indicative of the lack of maturity of a given branch of mathematics. Indeed, improving the standardization of mathematical notation for applied mathematics, which makes use of settled mathematical concepts, should have great benefit.

Perhaps it is due to the influence of the widespread use of computers that the last couple of decades have seen significant attempts to address the standardization of mathematical notation. Computers have matured from being primarily sophisticated numerical calculators, to where they now perform symbolic mathematical manipulations with computer algebra systems and automated theorem provers. These attempts at standardizing notation, at first confined to research communities [2-4] and then showing up in proprietary commercial products [5-7], have culminated in an effort to create a widespread public standard for the world-wide web [8].

2. Background

The Semantic Web is an idea conceived by the World-Wide Web Consortium (W3C) as an extension of the world-wide web. Tim Berners-Lee, inventor of Hyper-Text Markup Language (HTML) and the first web browser, is currently director of the W3C. Whereas HTML allowed

the creation and easy access and display of text-like documents, the semantic web consists of a set of constructs that will support the representation of layers of semantic descriptors, or metadata. These metadata, described in Extensible Markup Language (XML) [9] promise to lessen ambiguity and even support intelligent automated processing of documents on the web.

A variety of tools have arisen due to efforts of the W3C [10]. Recently, on February 9th, 2004, the W3C released the Resource Description Framework (RDF) and the OWL Web Ontology Language (OWL) as W3C Recommendations. RDF is used to represent information and to exchange knowledge in the Web. OWL is used to publish and share ontologies, supporting advanced Web search, software agents and knowledge management.

Another tool, RDF Schema describes how to use RDF to build RDF vocabularies. RDF Schema defines a basic vocabulary and conventions for use by Semantic Web applications.

The tools RDF, RDF Schema, OWL, etc. have been pulled together to deal with the challenges of representing all sorts of human knowledge. Since we are specifically interested in representing a small slice of that, i.e., specifying mathematical models, we are principally interested in an effort that comprises the current effort at standardizing mathematics.

Finally, another tools developed under the coordination of the W3C is the Mathematics Markup Language (MathML) [8] which we describe in more detail in this article.

3. MathML: Presentation vs. Content markup

Currently, for a large number of technical journals, the de-facto standard electronic format for submission of technical papers is LaTeX[10]. LaTex has been available for many years, on most operating systems, with many free versions, and above all, it has allowed authors to specify mathematical equations within the text of journal articles. One shortcoming of LaTeX with respect to mathematical content is that it is primarily oriented towards presentation, i.e., equation specifications amount to sophisticated typesetting specifications. A similar situation has

taken place with the world-wide web, where the hypertext markup language (HTML), which, while revolutionizing the communication occurring on computer networks is primarily oriented towards visual presentation.

The shortcomings of presentation-oriented specification may be explained by a simple example. Let us say that we want to write the following equation:

$$(1) \qquad x^i = \pi$$

In this equation a symbol, x, has a superscript, i, and is equated to the Greek symbol, π. This is relatively straightforward to represent with a variety of typesetting oriented applications. In reading this equation, we are still left with questions. What does x signify? Is the superscript an index, a label, or an exponent? Does the equality symbol represent assignment, as in a computer language? Is π used to represent the ratio of a circle's circumference to its diameter? These questions illustrate that, with a typeset equation there is no context and we must guess at the meaning of the terms as well as the meaning of the full mathematical sentence. We cannot be sure about the meaning of the symbols without supporting context, usually supplied in the non-standard, non-formal language of the embedding text.

As part of the semantic web effort, the W3C Math Activity is developing the MathML standard for representing mathematical knowledge. MathML, is comprised of two parts: one that focuses on presentation, called Presentation MathML, and one that focuses on content, called Content MathML.

While the presentation of mathematics is in itself important, we focus here on the representation of mathematical content, i.e., the semantic level of information conveyed in a mathematical statement. We do this because we believe that we should encourage technical authors not to worry so much about the appearance of their documents, but to focus on getting the right content. Authors shouldn't be concerned with a choice between using 18pt Times Roman, 12pt Times Italic for particular elements of a document. If they are required to think about these things, they waste their time with document design and create a lot of badly designed documents. It is better to leave document design to document designers, and to

let technical authors concern themselves with writing technical content.

What Content MathML provides is a standardized set of names and symbols for a variety of mathematical concepts as opposed to their visual representation. As an XML application, MathML may make use of a large set of Unicode characters to represent numbers and identifier symbols. Identifier symbols are strings of characters that are used as names. These are tagged by the token elements <ci></ci>, for content identifier symbols and <cn></cn>, for content numbers. For example, the number 64 is represented as

<cn>64</cn>

showing both the initiating and terminating tags. Another number, the mathematical constant, π, the ratio of a circle's circumference to its diameter, is represented as

<cn type="constant">π</cn>

This construct, using the ampersand and semi-colon, is used to express a set of MathML Entity Names. This is used in preference to using Unicode literals to represent a variety of symbols, resulting in a more human-readable representation. In some cases, such as with π, the default meaning of a constant number represented with such a symbol is the common meaning it holds. In other contexts π would usually be a readable representation for the Greek lower-case letter, π, not the transcendental.

To return to equation (1), the first identifier symbol, x, may be represented as

<ci>x</ci>

Note that this representation is of a scalar, by default, while the representation of a vector, **x**, would be

<ci type="vector">x</ci>

Note also that the typesetting or style of presentation is not expressed: a boldface or an arrow-above typographical representation of a vector may be expressed elsewhere, such as in a Presentation MathML annotation to the content or as a style-sheet definition.

The next concept for constructing mathematical expressions in Content MathML is the apply construct. The meaning of this construct is to apply a named operator to a list of arguments. For example, "equals" is represented by a symbol <eq/>, and the equation, x=64 would be represented by

Numerous operators are named in the standard. Another operator is , which allows one to express exponents of numbers or identifiers. We can now represent two possible meanings for equation (1), $x^i = \pi$, one, where the i-th power of x is equated to π

<apply>
        <cn type="constant">π</cn>
</apply>

or, alternatively, the i-th element of the vector **x** is equated to π

<apply>
        <apply>
                <ci type="vector">x</ci>
                <ci>i</ci>
        </apply>
        <cn type="constant">π</cn>
</apply>

This example illustrates some of the basic expressive capabilities of the Content MathML standard. As we continue, we will see the additional need to represent: complex numbers; multiplication, division, subtraction and addition; partial deriviatives, divergence, and gradient operations.

4. Physics-based Models

Of significant interest is the representation of mathematical expressions suitable for describing mathematical models of physical objects[11]. First, we see how MathML can help express more complex equations, such as partial differential equations.

As an example, we begin by trying to write a description of an acoustic wave field [12-13]. This begins with the wave equation describing the behavior of an acoustic pressure field to an impulsive acoustic point-source, i.e.,

$$(2)\ \nabla^2 G - \frac{\nabla\rho}{\rho} \bullet \nabla G + \frac{1}{c^2}\frac{\partial^2 G}{\partial t^2} = -\delta(r - r')\delta(t - t')$$

This equation is a starting point for our discussion. The function, G, sometimes called the "impulse response function", or "Green's function" represents the acoustic pressure at a point in space, **r,** at time t, due to an acoustic impulsive source at another point in space, **r**', at time t'. The current standard for representing MathML allows us to represent the following concepts directly. The Laplacian operator, ∇^2, is represented as <laplacian/>, or, alternatively as the divergence of the gradient, i.e.,

```
<apply>
        <divergence/>
        <apply>
                <gradient/>
                <ci type="function">G</ci>
        </apply>
</apply>
```

While we have here specified the type of G as "function", we could also have given it a type of "complex". The current MathML specification, i.e., MathML 2.0, 2nd edition, recognizes that multiple type specifiers, such as complex and function, may be simultaneously applicable. Future modifications to the standard are anticipated to support this [14] directly. In the meantime, users are advised to use a <semantics/> construct to create their own versions of these mathematical objects.

The next term in the wave equation, $\frac{\nabla\rho}{\rho} \bullet \nabla G$, may be represented as

```
<apply>
    <scalarproduct/>
    <apply>
```

```
        <divide/>
        <apply>
            <gradient/>
            <ci type="function">&rho</ci>
        </apply>
        <ci>&rho</ci>
    </apply>
    <apply>
        <gradient/>
        <ci type="function">G</ci>
    </apply
</apply>
```

The third term, $\frac{1}{c^2}\frac{\partial^2 G}{\partial t^2}$, may be represented as

```
<apply>
    <multiply/>
    <apply>
        <power/> <ci type="function">c</ci>
        <cn>-2</cn>
    </apply>
    <apply>
        <partialdiff/>
        <bvar><degree><cn>2</cn></degree>
            <ci>t</ci>
        </bvar>
        <degree><cn>2</cn></degree>
        <ci type="function">G</ci>
    </apply>
</apply>
```

We note that here we have specified the sound speed to be a function rather than a constant. One important mathematical concept not defined by the MathML strandard is that of the impulse function, or Dirac delta-function. MathML 2.0 gives a construct to define undefined concepts. Since initially we only need to unambiguously refer to the Dirac delta function, rather than make use of its properties to perform some evaluation, the first thing we need is a name. We would prefer <diracdelta/>, but instead must use the MathML <csymbol> construct to create a representation of the concept. We can supply a universal resource locator, or URL, to provide a definition that we write ourselves. Considering this, the left-hand side of the wave equation, $-\delta(r - r')\delta(t - t')$, may then be represented as

```
<apply>
    <product/>
    <cn>-1</cn>
    <apply>
        <csymbol encoding="text"
            definitionURL="http://www.ait.nrl.na
            vy.mil/missingmath/diracdelta.htm>
```

```
        <msub><mi>&delta;</mi></msub>
    </csymbol>
    <apply>
        <minus/>
        <ci type="vector">r</ci>
        <ci type="vector">r&apos;</ci>
    </apply>
</apply>
<apply>
    <csymbol encoding="text"
      definitionURL="http://www.ait.nrl.na
      vy.mil/missingmath/diracdelta.htm>
        <msub><mi>&delta;</mi></msub>
    </csymbol>
    <apply>
        <minus/>
        <ci>t</ci>
        <ci>t&apos;</ci>
    </apply>
</apply>
</apply>
</apply>
```

In substituting our own symbol for the dirac delta function we have specified its appearance, using the <msub> and <mi> tags, but we have not here specified the underlying mathematical properties of the symbol. For example we know that

$$f(0) = \int_{-|\varepsilon|}^{|\varepsilon|} \delta(x) f(x) dx \quad , \quad \varepsilon \neq 0$$

While it is best that the semantics be specified, there is currently no standard telling us how to do so: it seems that an empty definitionURL will not affect any automated interpretation of the symbol. The primary value that the <csymbol> construct gives us is in user-defined labels for concepts undefined in the MathML specification. We may also consider that another XML application for describing mathematical content, OpenMath[3], provides a little more help in this regard by supporting user-developed content-dictionaries. OpenMath constructs may be used within the same XML-based document as MathML descriptions.

5. Where is the Physics?

While the above example is taken from an equation representing physical phenomena, it is still only a mathematical equation: the physical meaning is not in the XML-based description. Some concepts that need to be expressed in order to state the physical meaning are as follows. We

need the notion of a space-time, where physical space-time is an instance of a type of mathematical metric-space. We need the notion of a class, suggestively named Physical_Object, that allows us to attribute a name, type, and set of measurable physical properties to objects that are modeled. For example, how would we tag the MathML specification of equation (2) above so as to indicate that we are modeling the propagation of acoustic waves in the fluid-body representation of an object we refer to as "the ocean"? How do we state that the position and time symbols, which refer to Newtonian space-time, have the properties of elements in a Euclidean metric-space?

The main point here is that when we create math-based models, we use symbols that are loaded with meaning. Content MathML is a specification that allows us to describe much of the mathematical properties of those symbols. What it does not provide is a way to describe how we simultaneously use those same symbols to represent objects in models of reality. In order to do that we must develop associated standards. For example, a document [15] describing how the representation of physical units may be implemented within MathML is available with other MathML documentation, but it is expressly stated that this is not intended to be a part of the MathML standard.

6. Summary

We have described how we can begin to document mathematical models, using a standard, Content MathML, that focuses on specifying the mathematical content of those models. We have indicated that work still needs to be done to clarify how that mathematical content may need to be augmented with modeling constructs that are specific to a mathematically described scientific content such as physics-based models.

7. Acknowledgments

8. References

[1] "Vectors Versus Quaternions – The Letters of Nature", Alfred M. Bork, American Journal of Physics 34 (1966) 202-211

[2] "An Ontology for Engineering Mathematics," Thomas R. Gruber and Gregory R. Olsen, in "Fourth International Conference on Principles of Knowledge Representation and Reasoning", Morgan Kaufmann, 1994.

[3] http://www.openmath.org/cocoon/openmath//index.html

[4] http://mizar.org/project/

[5] The Mathematica Book, Fifth Edition, Stephen Wolfram, Wolfram Media (2003)

[6] "The MAPLE Book", Frank Garvan; Chapman & Hall/CRC (2001)

[7] MuPAD Tutorial, Jurgen Gerhard, WalterOevel, Frank Postel, Stefan Wehmeier; Springer (2000)

[8] http://www.w3.org/Math/

[9] http://www.w3.org/XML/

[10] http://www.latex-project.org/

[11] Collins, Joseph B., and Clark, Doug "Towards an Ontology of Physics", In *Proceedings of the European Simulation Interoperability Workshop*, July 2004, 04E-SIW-044.

[12] "The FeyRay Gaussian Beam Underwater Acoustic Propagation Model", Terry L. Foreman, unpublished technical report.

[13] "Expanded Sonar Environmental Acoustic Effects Server Capabilities for Fleet Battle Experiments", Terry Foreman and Dan Speicher, Proceedings of the Fall Simulation Interoperability Workshop, September 2003, paper #03F-SIW-062

[14] http://www.w3.org/TR/MathML2/chapter4.html#contm.ci

[15] http://www.w3.org/TR/mathml-units/